Конвейерные схемы, вычисляющие несколько выражений

- Южный федеральный университет, мехмат
-
- ***Б.Я. Штейнберг,*** *А.П. Баглий, Ж.М. Петрова, О.Б. Штейнберг*

SECR

Открытая распараллеливающая система

Ускорители на ПЛИС

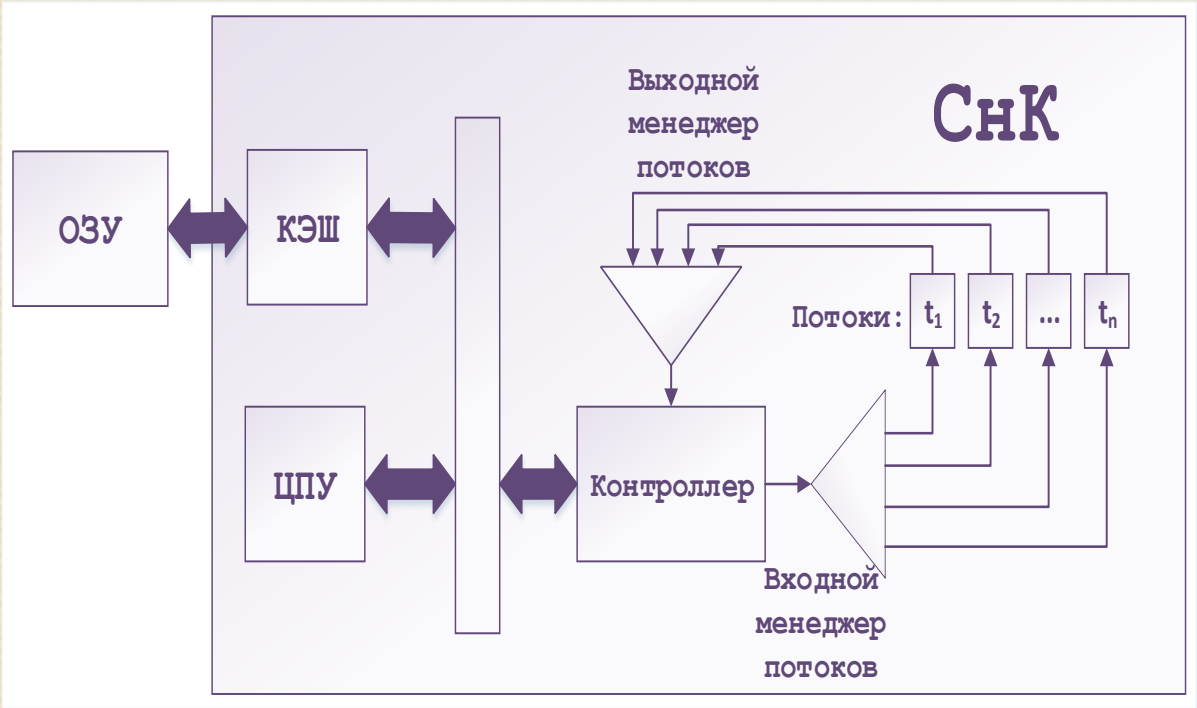
- Система Maxwell фирмы FHPCA (FPGA High Performance Computing Alliance) состоит из 32 «Блэйд-серверов», (один процессор Xeon и две PCI-E карты на ПЛИС Virtex4) для ряда задач, например решения дифференциальных уравнений методом Монте-Карло, показала **250-1000-кратный прирост производительности** по сравнению с аналогичной системой, не применяющей ПЛИС.
- Карта XILINX ALVEO предназначена для увеличения производительности дата центров, суперкомпьютеров и демонстрирует производительность в 90 раз большую по сравнению с CPU и в 4 раза по сравнению с GPU.

Развитие новых архитектур сдерживается отсутствием инструментального ПО

- ПЛИСы уже несколько десятилетий демонстрируют конкурентные преимущества в производительности и энергопотреблении, но занимают значительно меньшую часть рынка микросхем, чем могли бы, поскольку не имеют хорошего инструментального программного обеспечения.

ОСОБЕННОСТИ КОМПИЛЯТОРА НА ПРОГРАММИРУЕМУЮ АРХИТЕКТУРУ (Ю. МИХАЙЛУЦ & К°)

- Низкое энергопотребление
- Высокая производительность
- \+ Высокая скорость обмена ЦПУ и ПЛИС.
- \+ Не требуется сложный протокол обмена данными.
- Масштабируемость ограничена логическими ресурсами кристалла.



Особенности компилятора:

1)содержит конвертор C2HDL;

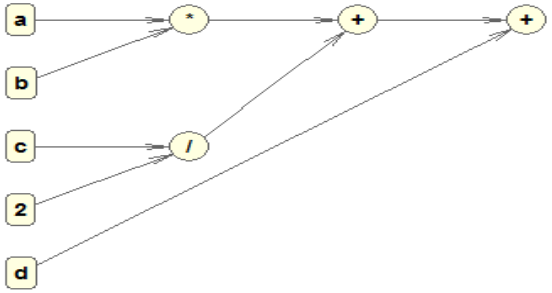
2) содержит генератор драйверов;

3) ПЛИС поддерживает только часть кода;

4) специальные высокоуровневые оптимизации.

Вычислительный конвейер

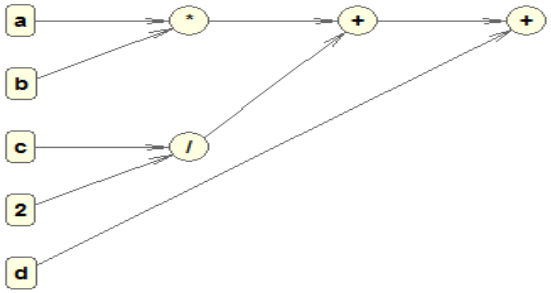
- *Пример. для вычисления кода* $a * b + c/2 + d$
- **Можно создать схему (граф) вычислений**



- Этот же граф вычислений может вычислить и выражение $a * y + x/2 + d$

Вычисление одинаковых выражений

- *Конвейер, вычисляющий выражение* $a * b + c/2 + d$
- может использоваться для вычисления тела цикла
- **For i=1 to N do**
- **X[i] = a[i]*b[i]+c[i]/2+d[i]**



- При этом, все операции будут работать одновременно

Реконфигурируемый конвейер

- Для вычисления некоторого другого выражения
- $a * y/z + x * u * v/2$
- нужен другой конвейер.
- Ограниченные ресурсы ПЛИС могут содержать несколько конвейеров, но не очень больших и не очень много.

- **Программируемые архитектуры (ПЛИС) быстро вычисляют, но долго перепрограммируются**

Перепрограммирование конвейера

- Программируемые архитектуры (ПЛИС) быстро вычисляют, но долго перепрограммируются
- И.И. Левин (НИИ МВС ЮФУ) перепрограммирует только связи между вычислительными устройствами (сумматорами, умножителями,…)
- **K. Bondalapati & V. Prasana** предлагают алгоритм минимизации перепрограммирований конвейера для вычисления кода.

Конвейер, покрывающий несколько других конвейеров

- В данной работе предлагается строить конвейер, на котором можно вычислять несколько исходных выражений.

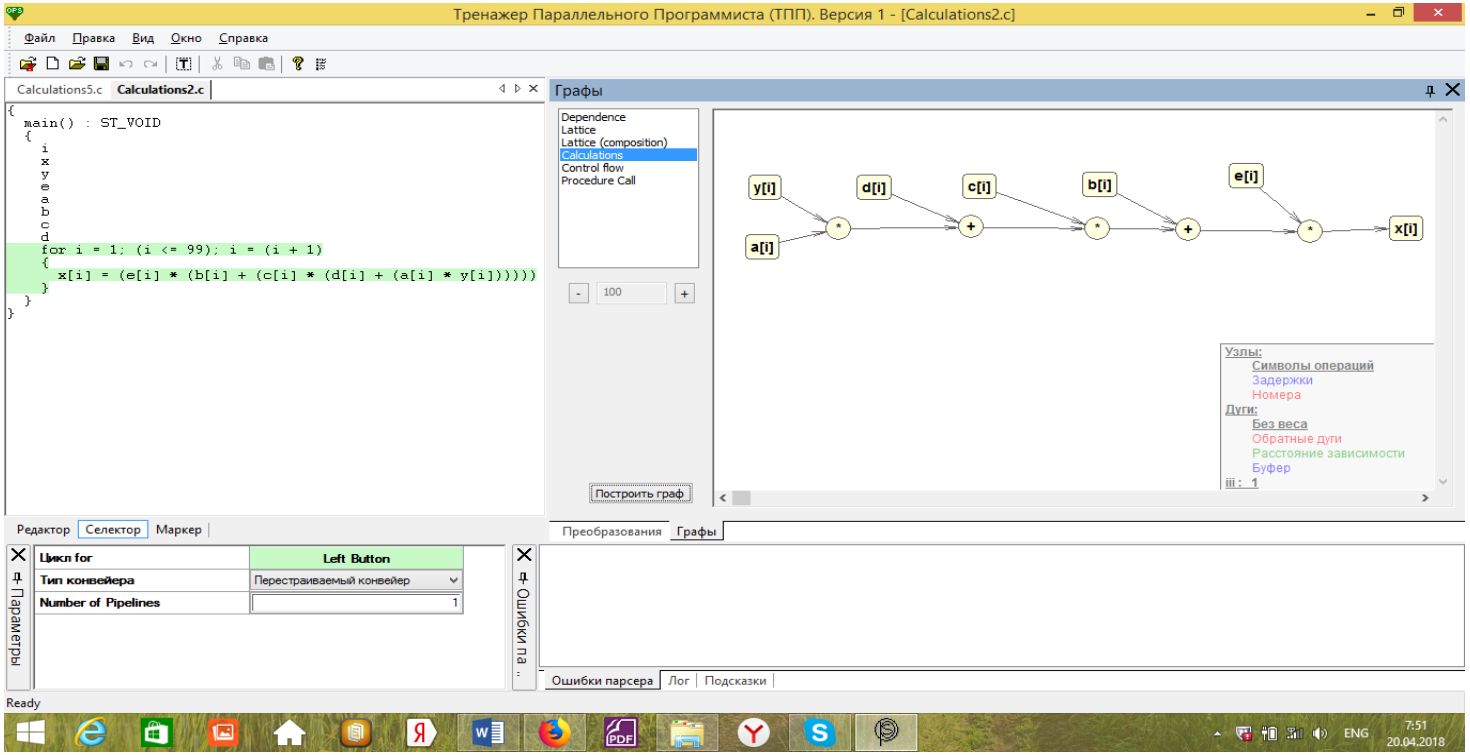
- Предлагаемый конвейер может оказаться сложнее, чем каждый конвейер, построенный в соответствии с каждым входным выражением. Но необходимые ресурсы ПЛИС для построения предлагаемого конвейера меньше, чем совокупные ресурсы для конвейеров всех входных выражений.

Конвейер, покрывающий несколько других конвейеров

- Пример. Для двух выражений
- $(x * x + a * x)/(x + c)$
- $(x * x + b)/(x + c)$
-
- Создадим новое выражение
- $(x * x + a * x + b)/(x + c)$

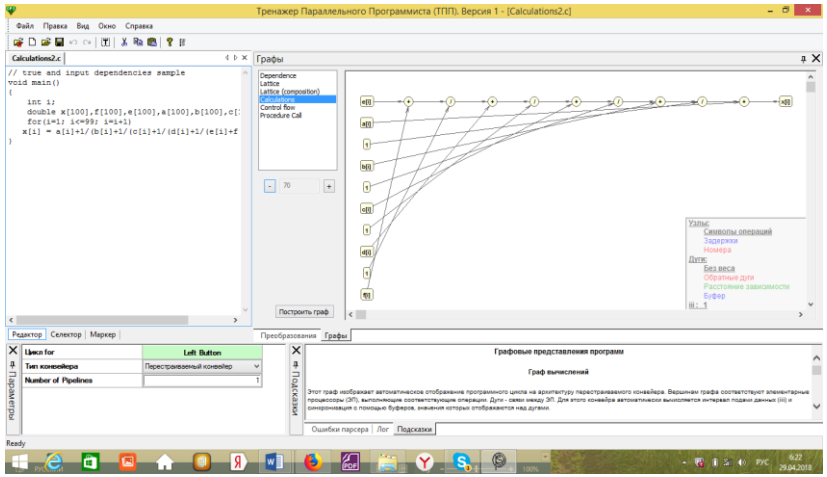
- Исходные выражения могут быть получены фиксацией значений переменных $b = o$ и $a = o$ соответственно, и могут вычисляться одним конвейерным вычислителем.

Граф вычислений – промежуточное представление реконфигурируемого конвейера в компиляторе



- Граф вычислений, построенный автоматически в Оптимизирующей распараллеливающей системе (ОРС)

Выражения, у которых граф вычислений – простой путь



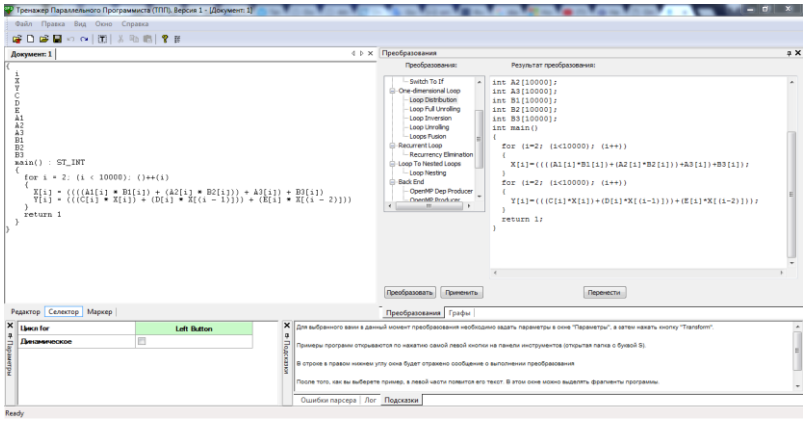
- Граф вычислений цепной дроби – простой путь.
- Граф вычислений схемы Горнера можно рассматривать, как простой путь.

Применение алгоритмов выравнивания последовательностей

- Если исходные выражения представляют собой простой путь, для построения покрывающего выражения можно использовать известные алгоритмы выравнивания строк.
- Такие алгоритмы используются в биоинформатике для выравнивания нуклеотидных последовательностей.
- Для парного выравнивания можно использовать алгоритм Нидлмана-Вунша. Множественные выравнивания имеют очень высокую алгоритмическую сложность и могут строится на основе парных.
- Для общего случая (в процессе) алгоритмы имеют сложный переборный характер, причем перебор деревьев ожидается не простым..

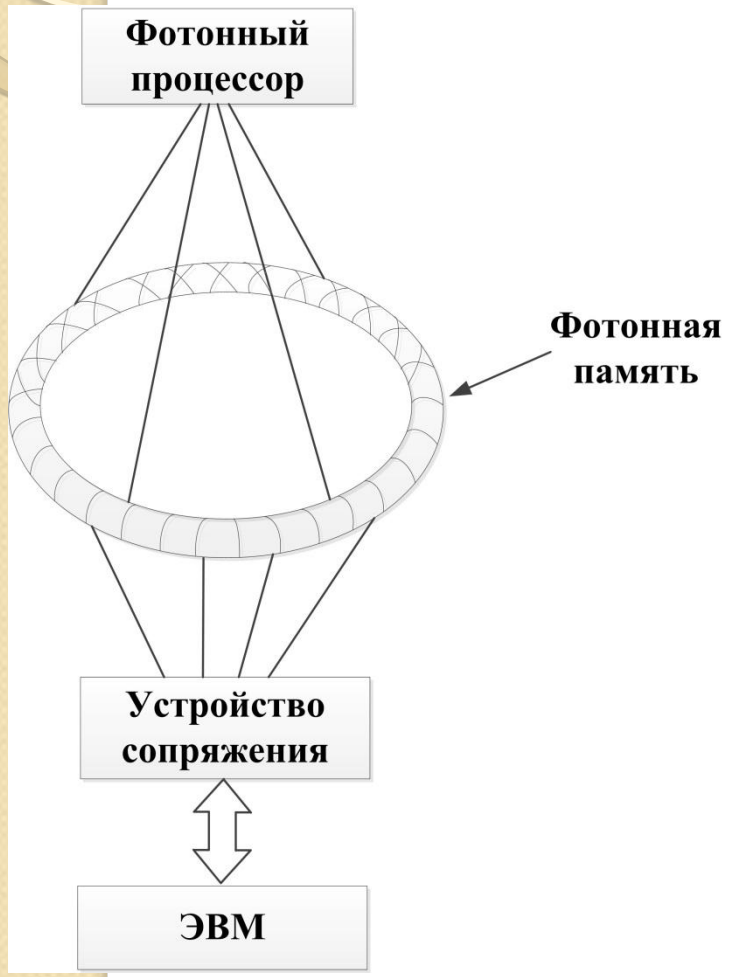
Большое тело конвейеризуемого цикла

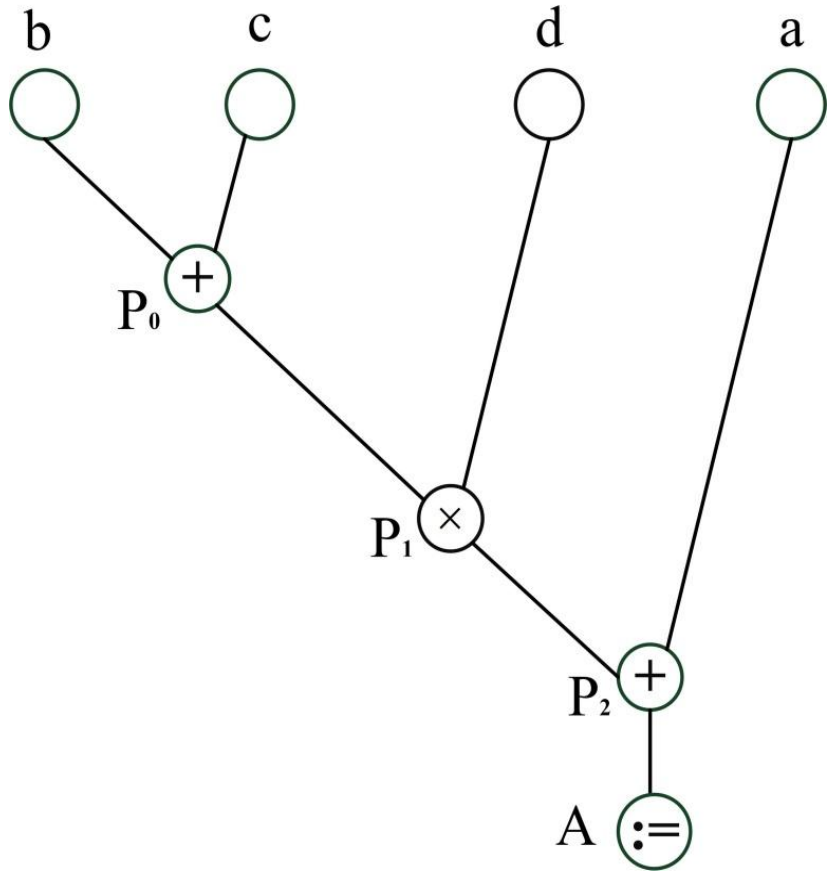
- Может случиться, что на ПЛИС недостаточно ресурсов для создания конвейера одного исходного конвейеризуемого цикла.



- Скриншот преобразования «разбиение циклов» в ОРС.
- Можно применить преобразование «разбиение цикла» и из одного большого исходного цикла получить несколько более мелких циклов, для которых искать покрывающий цикл.

Фотонный компьютер (рисунки С.А. Степаненко): основа программы как и для программируемого конвейера – граф вычислений





Сингулярные гнезда циклов для передачи на ускорители

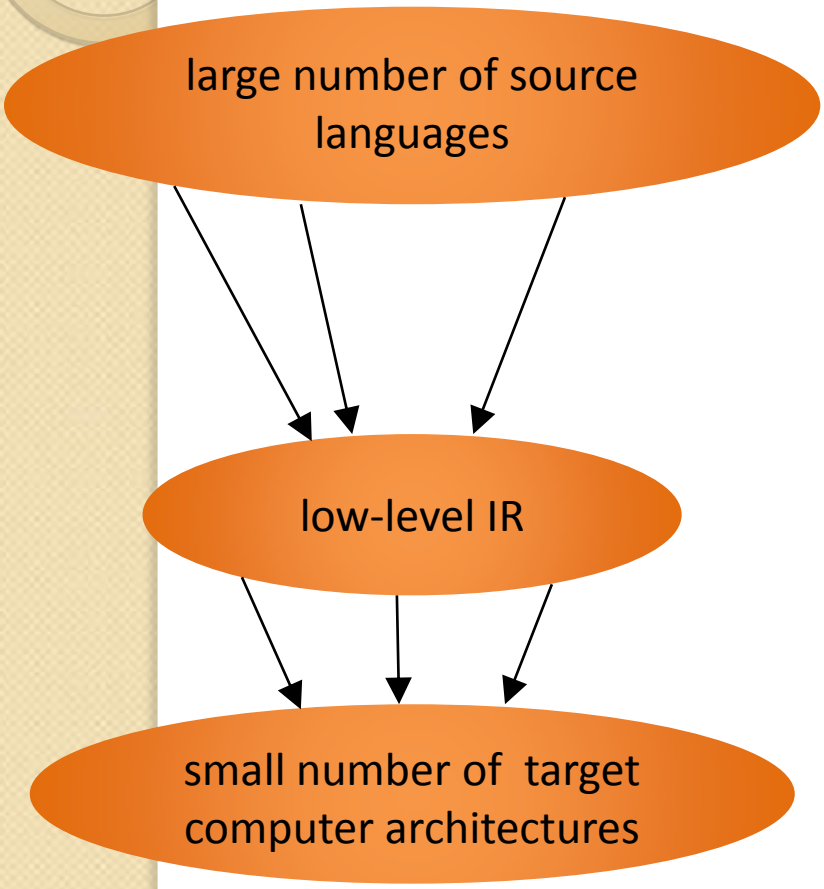
- Поскольку доступ к данным в оперативной памяти на современных компьютерах намного дольше вычислительных операций, на ускорители передаются части программы, у которых много вычислений с небольших объемом данных (которые должны поместиться на локальной памяти микросхемы ускорителя, в данном случае, ПЛИС). Такими фрагментами кода могут быть такие гнезда циклов, у которых количество циклов гнезда больше размерности любого входящего в гнездо массива. Такие гнезда можно называть сингулярными (вырожденными). Такими являются гнезда циклов алгоритма перемножения матриц, алгоритма Флойда-Уоршала и др. В ОРС реализован поиск таких участков кода .

Преимущества высокоуровневого ВП компилятора по сравнению с регистровым

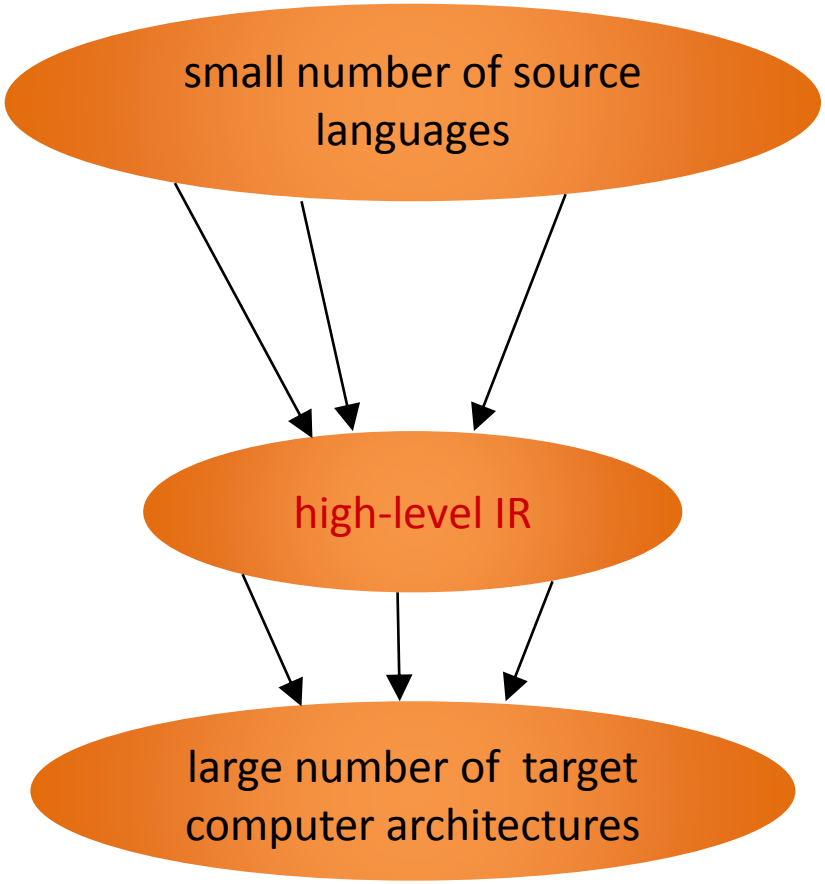
1. При диалоговом анализе программы пользователь лучше узнает свой код, который ему возвращается преобразованным из высокоуровневого ВП, чем из регистрового.
2. Легче анализировать код для блочного размещения данных в общей памяти.
3. Легче анализировать код для блочно-аффинных размещений данных с перекрытиями в распределенной памяти.
4. Легче генерировать HDL-описание конвейерного вычислителя

Low/High Level IR Comparison

low-level IR



high-level IR



Нестандартные функции компилятора

- Поиск сингулярных гнезд циклов (много вычислений с малым числом данных)
- Диалоговое преобразование программ к блочным вычислениям (тайлинг)
- Минимизация перенастроек конвейера (поиск покрывающего дерева для нескольких деревьев)
- Retming для оптимизации буферной памяти для синхронизации конвейера
- Распараллеливание рекуррентных вычислений
- Диалоговый режим оптимизирующей компиляции с символьным анализом и уточнением зависимостей, с рекомендациями по оптимизации.
- Блочные размещения массивов в оперативной памяти
- Блочно-аффинные размещения массивов в распределенной памяти

http://ops.rsu.ru